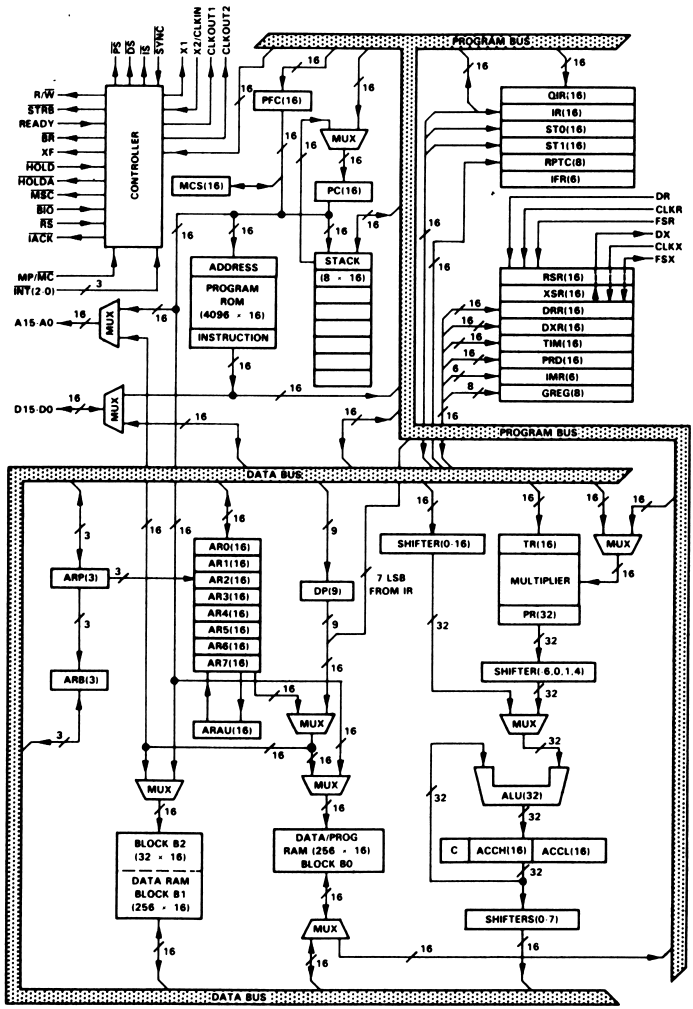
programs can execute at full speed from this memory space. Programs may also be downloaded from slow external memory to on-chip RAM for full-speed operation. The VLSI implementation of the TMS320C25 incorporates all of these features as well as many others such as a hardware timer, serial port, and block data transfer capabilities.

A functional block diagram of the TMS320C25, shown in Fig. 5, outlines the principal blocks and data paths within



LEGEND:

ACCH	Accumulator high	IFR	Interrupt flag register	PC	Program counter
ACCL	Accumulator low	IMR	Interrupt mask register	PFC	Prefetch counter
ALU	Arithmetic logic unit	IR	Instruction register	RPTC	Repeat instruction counter
ARAU	Auxiliary register arithmetic unit	MCS	Microcall stack	GREG	Global memory allocation register
ARB	Auxiliary register pointer buffer	OIR	Queue instruction register	RSR	Serial port receive shift register
ARP	Auxiliary register pointer	PR	Product register	XSR	Serial port transmit shift register
DP	Data memory page pointer	PRD	Period register for timer	AR0-AR7	Auxiliary registers
DRR	Serial port data receive register	TIM	Timer	ST0,ST1	Status registers
DXR	Serial port data transmit register	TR	Temporary register		

Fig. 5. TMS320C25 functional block diagram.

the processor. The diagram also shows all of the TMS320C25 interface pins.

In the following architectural discussions on the memory, central arithmetic logic unit, hardware multiplier, control operations, serial port, and I/O interface, please refer to the block diagram shown in Fig. 5.

Memory Allocation: The TMS320C25 provides a total of 4K 16-bit words of on-chip program ROM and 544 16-bit words of on-chip data RAM. The RAM is divided into three separate Blocks (B0, B1, and B2). Of the 544 words, 256 words (block B0) are configurable as either data or program memory by CNFD (configure data memory) or CNFP (configure program memory) instructions provided for that purpose; 288 words (blocks B1 and B2) are always data memory. A data memory size of 544 words allows the TMS320C25 to handle a data array of 512 words while still leaving 32 locations for intermediate storage. The TMS320C25 provides 64K words of off-chip directly addressable data memory space as well as a 64K-word off-chip program memory space.

A register file containing eight Auxiliary Registers (AR0–AR7), which are used for indirect addressing of data memory and for temporary storage, increase the flexibility and efficiency of the device. These registers may be either directly addressed by an instruction or indirectly addressed by a 3-bit Auxiliary Register Pointer (ARP). The auxiliary registers and the ARP may be loaded from either data memory or by an immediate operand defined in the instruction. The contents of these registers may also be stored into data memory. The auxiliary register file is connected to the Auxiliary Register Arithmetic Unit (ARAU). Using the ARAU accessing tables of information does not require the CALU for address manipulation, thus freeing it for other operations.

Central Arithmetic Logic Unit (CALU): The CALU contains a 16-bit scaling shifter, a 16 × 16-bit parallel multiplier, a 32-bit Arithmetic Logic Unit (ALU), and a 32-bit accumulator. The scaling shifter has a 16-bit input connected to the data bus and a 32-bit output connected to the ALU. This shifter produces a left-shift of 0 to 16 bits on the input data, as programmed in the instruction. Additional shifters at the outputs of both the accumulator and the multiplier are suitable for numerical scaling, bit extraction, extended-precision arithmetic, and overflow prevention.

The following steps occur in the implementation of a typical ALU instruction:

1) Data are fetched from the RAM on the data bus.
2) Data are passed through the scaling shifter and the ALU where the arithmetic is performed.
3) The result is moved into the accumulator.

The 32-bit accumulator is split into two 16-bit segments for storage in data memory: ACCH (accumulator high) and ACCL (accumulator low). The accumulator has a carry bit to facilitate multiple-precision arithmetic for both addition and subtract instructions.

Hardware Multiplier: The TMS320C25 utilizes a 16 × 16-bit hardware multiplier, which is capable of computing a 32-bit product during every machine cycle. Two registers are associated with the multiplier:

- a 16-bit Temporary Register (TR) that holds one of the operands for the multiplier, and
- a 32-bit Product Register (PR) that holds the product.

The output of the product register can be left-shifted 1 or 4 bits. This is useful for implementing fractional arithmetic or justifying fractional products. The output of the PR can also be right-shifted 6 bits to enable the execution of up to 128 consecutive multiple/accumulates without overflow. An unsigned multiply (MPYU) instruction facilitates extended-precision multiplication.

I/O Interface: The TMS320C25 I/O space consists of 16 input and 16 output ports. These ports provide the full 16-bit parallel I/O interface via the data bus on the device. A single input (IN) or output (OUT) operation typically takes two cycles; however, when used with the repeat counter, the operation becomes single-cycle. I/O devices are mapped into the I/O address space using the processor's external address and data buses in the same manner as memory-mapped devices. Interfacing to memory and I/O devices of varying speeds is accomplished by using the READY line.

A Direct Memory Access (DMA) to external program/data memory is also supported. Another processor can take complete control of the TMS320C25's external memory by asserting HOLD low, causing the TMS320C25 to place its address, data, and control lines in the high-impedance state. Signaling between the external processor and the TMS320C25 can be performed using interrupts. Two modes of DMA are available on the device. In the first, execution is suspended during assertion of HOLD. In the second "concurrent DMA" mode, the TMS320C25 continues to execute its program while operating from internal RAM or ROM, thus greatly increasing throughput in data-intensive applications.

TMS320C25 Software

The majority of the TMS320C25 instructions (97 out of 133) are executed in a single instruction cycle. Of the 36 instructions that require additional cycles of execution, 21 involve branches, calls, and returns that result in a reload of the program counter and a break in the execution pipeline. Another seven of the instructions are two-word, long-immediate instructions. The remaining eight instructions support I/O, transfers of data between memory spaces, or provide for additional parallel operation in the processor. Furthermore, these eight instructions (IN, OUT, BLKD, BLKP, TBLR, TBLW, MAC, and MACD) become single-cycle when used in conjunction with the repeat counter. The functional performance of the instructions exploits the parallelism of the processor, allowing complex and/or numerically intensive computations to be implemented in relatively few instructions.

Addressing Modes: Since most of the instructions are coded in a single 16-bit word, most instructions can be executed in a single cycle. Three memory addressing modes are available with the instruction set: direct, indirect, and immediate addressing. Both direct and indirect addressing are used to access data memory. Immediate addressing uses the contents of the memory addressed by the program counter.

When using direct addressing, 7 bits of the instruction word are concatenated with the 9 bits of the data memory page pointer (DP) to form the 16-bit data memory address. With a 128-word page length, the DP register points to one of 512 possible data memory pages to obtain a 64K total data memory space. Indirect addressing is provided by the aux-

iliary registers (AR0–AR7). The seven types of indirect addressing are shown in Table 4. Bit-reversed indexed addressing modes allow efficient I/O to be performed for the resequencing of data points in a radix-2 FFT program.

Table 4 Addressing Modes of the TMS320C25

Addressing Mode	Operation
OP A	direct addressing
OP * (,NARP)	indirect; no change to AR.
OP *+(,NARP)	indirect; current AR is incremented.
OP *−(,NARP)	indirect; current AR is decremented.
OP *0+(,NARP)	indirect; AR0 is added to current AR.
OP *0−(,NARP)	indirect; AR0 is subtracted from current AR.
OP *BR0+(,NARP)	indirect; AR0 is added to current AR (with reverse carry propagation).
OP *BR0−(,NARP)	indirect; AR0 is subtracted from current AR (with reverse carry propagation).

Note: The optional NARP field specifies a new value of the ARP.

TMS320C25 System Configurations

The flexibility of the TMS320C25 allows systems configurations to satisfy a wide range of application requirements [16]. The TMS320C25 can be used in the following configurations:

- a stand-alone system (a single processor using 4K words of on-chip ROM and 544 words of on-chip RAM),
- parallel multiprocessing systems with shared global data memory, or
- host/peripheral coprocessing using interface control signals.

A minimal processing system is shown in Fig. 6 using external data RAM and PROM/EPROM. Parallel multiprocessing and host/peripheral coprocessing systems can be designed by taking advantage of the TMS320C25's direct memory access and global memory configuration capabilities.

In some digital processing tasks, the algorithm being implemented can be divided into sections with a distinct processor dedicated to each section. In this case, the first and second processors may share global data memory, as well as the second and third, the third and fourth, etc. Arbitration logic may be required to determine which section of the algorithm is executing and which processor has access to the global memory. With multiple processors dedicated to distinct sections of the algorithm, throughput can be increased via pipelined execution. The TMS320C25 is capable of allocating up to 32K words of data memory as global memory for multiprocessing applications.

The Third Generation of the TMS320 Family

The TMS320C30 [26]–[27] is Texas Instruments third-generation member of the TMS320 family of compatible digital signal processors. With a computational rate of 33 MFLOPS (million floating-point operations per second), the TMS320C30 far exceeds the performance of any programmable DSP available today. Total system performance has been maximized through internal parallelism, more than twenty-four thousand bytes of on-chip memory, single-cycle floating-point operations, and concurrent I/O. The total system cost is minimized with on-chip memory and on-chip peripherals such as timers and serial ports. Finally, the user's system design time is dramatically reduced with the availability of the floating-point operations, general-purpose instructions and features, and quality development tools.

The TMS320C30 provides the user with a level of performance that, at one time, was the exclusive domain of supercomputers. The strong architectural emphasis of providing a low-cost system solution to demanding arithmetic algorithms has resulted in the architecture shown in Fig. 7.

The key features of the TMS320C30 [26], [27] are as follows:

- 60-ns single-cycle execution time, 1-μm CMOS.
- Two 1K × 32-bit single-cycle dual-access RAM blocks.
- One 4K × 32-bit single-cycle dual-access ROM block.
- 64 × 32-bit instruction cache.
- 32-bit instruction and data words, 24-bit addresses.
- 32/40-bit floating-point and integer multiplier.
- 32/40-bit floating-point, integer, and logical ALU.
- 32-bit barrel shifter.
- Eight extended-precision registers.
- Two address-generators with eight auxiliary registers.
- On-chip Direct Memory Access (DMA) controller for concurrent I/O and CPU operation.
- Peripheral bus and modules for easy customization.
- High-level language support.
- Interlocked instructions for multiprocessing support.
- Zero overhead loops and single-cycle branches.

The architecture of the TMS320C30 is targeted at 60-ns and faster cycle times. To achieve such high-performance

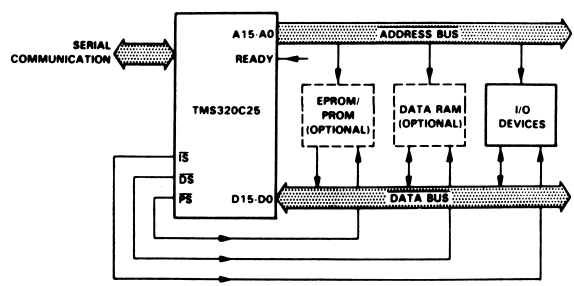


Fig. 6. Minimal processing system with external data RAM and PROM/EPROM.

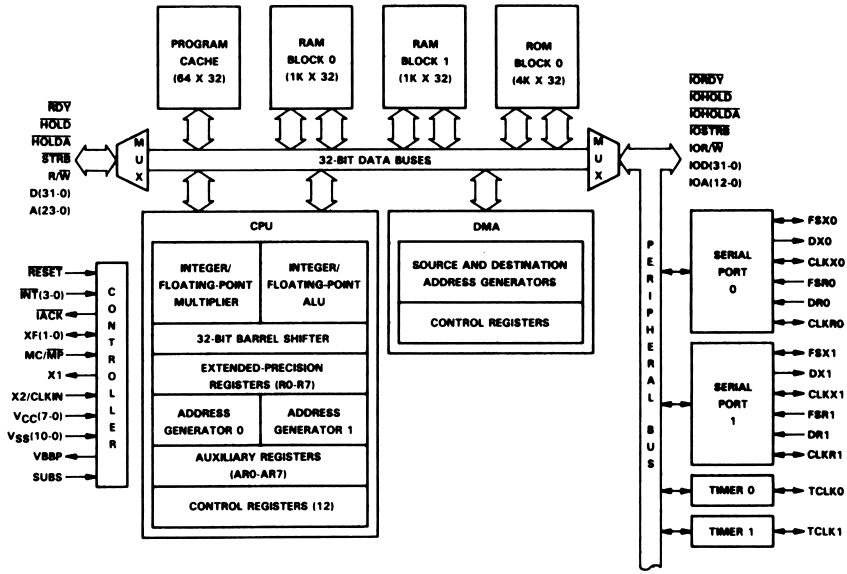


Fig. 7. TMS320C30 functional block diagram.

goals while still providing low-cost system solutions, the TMS320C30 is designed using Texas Instruments state-of-the-art 1-μm CMOS process. The TMS320C30's high system performance is achieved through a high degree of parallelism, the accuracy and precision of its floating-point units, its on-chip DMA controller that supports concurrent I/O, and its general-purpose features. At the heart of the architecture is the Central Processing Unit (CPU).

The CPU

The CPU consists of the following elements: floating-point/integer multiplier; ALU for performing floating-point, integer, and logical operations; auxiliary register arithmetic units; supporting register file, and associated buses. The multiplier of the CPU performs floating-point and integer multiplication. When performing floating-point multiplication, the inputs are 32-bit floating-point numbers, and the result is a 40-bit floating-point number. When performing integer multiplication, the input data is 24 bits and yields a 32-bit result. The ALU performs 32-bit integer, 32-bit logical, and 40-bit floating-point operations. Results of the multiplier and the ALU are always maintained in 32-bit integer or 40-bit floating-point formats. The TMS320C30 has the ability to perform, in a single cycle, parallel multiplies and adds (subtracts) on integer or floating-point data. It is this ability to perform floating-point multiplies and adds (subtracts) in a single cycle which give the TMS320C30 its peak computational rate of 33 MFLOPS.

Floating-point operations provide the user with a convenient and virtually trouble-free means of performing computations while maintaining accuracy and precision. The TMS320C30 implementation of floating-point arithmetic allows for floating-point operations at integer speeds. The floating-point capability allows the user to ignore, to a large extent, problems with overflow, operand alignment, and other burdensome tasks common to integer operations.

The register file contains 28 registers, which may be operated upon by the multiplier and ALU. The first eight of these registers (R0–R7) are the extended-precision registers, which support operations on 40-bit floating-point numbers and 32-bit integers.

The next eight registers (AR0–AR7) are the auxiliary registers, whose primary function is related to the generation of addresses. However, they also may be used as general-purpose 32-bit registers. Two auxiliary register arithmetic units (ARAU0 and ARAU1) can generate two addresses in a single cycle. The ARAUs operate in parallel with the multiplier and ALU. They support addressing with displacements, index registers (IR0 and IR1), and circular and bit-reversed addressing.

The remaining registers support a variety of system functions: addressing, stack management, processor status, block repeat, and interrupts.

Data Organization

Two integer formats are supported on the TMS320C30: a 16-bit format used for immediate integer operands and a 32-bit single-precision integer format.

Two unsigned-integer formats are available: a 16-bit format for immediate unsigned-integer operands and a 32-bit single-precision unsigned-integer format.

The three floating-point formats are assumed to be normalized, thus providing an extra bit of precision. The first